North South University

Department of Computer Science and Engineering

Senior Design Project Report - Fall 2022

SHONGKHEPON (সংক্ষেপণ)

Summarization Highlighting Only Necessary Graphics from videos using K-means in Hybrid with Euclidean distance and PCA Optimizing Neural networks

Submitted by:

Mahira Ibnath Joytu 1831232642

Tasnim Islam Plabon 1911088642

Labiba Binte Ismail 1911846642

Supervisor

Dr. Mohammad Ashrafuzzaman Khan (AzK)

Assistant Professor, Department of Electrical and Computer Engineering

North South University, Dhaka, Bangladesh

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at North South University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mahira Ibnath Joytu

1831232642

Tasnim Islam Plabon

1911088642

Labiba Binte Ismail

1911846642

Approval

This senior design capstone project titled "SHONGKHEPON: Summarization Highlighting Only Necessary Graphics from videos using K-means in Hybrid with Euclidean distance and PCA Optimizing Neural networks" submitted by Mahira Ibnath Joytu (1831232642), Tasnim Islam Plabon (1911088642), Labiba Binte Ismail (1911846642) has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on Fall 2022.

Approved By:

Supervisor

Dr. Mohammad Ashrafuzzaman Khan

Asst. Professor, Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

Department Chair

Dr. Rajesh Palit

Professor, Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

Acknowledgement

Dr. Mohammad Ashrafuzzaman Khan, our honorable project supervisor, provided unwavering support, inspiration, and direction that was essential to the project's success. We would like to sincerely thank our distinguished supervisor, Dr. Mohammad Ashrafuzzaman Khan, for his careful direction, insightful input, and patience and motivation to see this endeavor through to completion. We occasionally had severe doubts about finishing this job, but Dr. Ashraf consistently gave us advice, pointed us in the right direction, and provided us with workable answers. We could not have completed our project in its current form without his assistance. Finally, we would want to express our gratitude to everyone who helped us and offered advice in order for this project to be completed.

Abstract

Nowadays, people are very busy and need help to hold their attention span. They need more time to watch an entire film, web film, or series. Even 10-minute videos are sometimes too long for us. Short videos have been trendy nowadays because of how quickly you can glance at your screen and gain information in no time. And for that reason, we usually miss out on the little entertainment we get. Even avid movie lovers need more time to follow their series. Finding significant or instructive portions of the original video involves understanding its content, which makes this a difficult task. Moreover, the variety of Internet videos subjects—from family videos to documentaries—makes it difficult to summarize them because prior knowledge is seldom available. Hence, we found ways to automate this process with machine learning-based video summarization techniques. And we came out with a video summary method that lets you shorten videos from an input video file, highlighting only the essential moments and removing repetitive scenes. In terms of key frame extraction, there can be different biases to determine which is essential. To solve that problem, we tried to use an unsupervised learning algorithm, K-means clustering. Clustering is frequently used when encountering similar characteristics or activities within a frame. It also helps to eliminate those frames that have periodic trends. Before that, we applied Principal Component Analysis (PCA) to remove noise and redundancy while extracting keyframes.

Keywords: Keyframe, K-means clustering, CNN, PCA

Table of Contents

List of Figures

Chapter 1

Introduction

It's become quite challenging to effectively search among the millions of videos on the internet due to the tremendous expansion of video content. Users are frequently confused by the enormous number of videos given by search engines like Google when conducting an event query. The user experience can decline and be time-consuming when exploring such findings.

The technique of condensing a video by picking keyframes or segments that best convey its major ideas is known as video summarization, and AI is increasingly important in this process. One of the essential applications for summarization is the capacity to determine the level of interest in a piece of content. How many people view a complete video can be predicted by its flashcard synopsis. Also, how many people will click on a video to play depends heavily on even just one thumbnail. Video summary is essential for compelling viewing of the content and adjusting the duration of videos for different platforms, such as Instagram, Facebook, etc.

Deep learning has recently made significant strides in processing images, and the accuracy of AI's understanding of an image's context has improved exponentially. Videos can also be understood using similar methods, although this is a much more difficult task. Videos are multi-dimensional and include voice, motion, and a time-series dimension in addition to being simply a collection of numerous frames or

images. Each of these factors is important for comprehending a video, and different factors may be important depending on the target audience for the summarization.

1.1 Thoughts behind the Model

Our main idea behind the model was to make something that can summarize lengthy videos into shorter ones for quick access and viewing by implementing deep learning methods. Sitting and immersing ourselves in the entertainment world is becoming more challenging daily in our busy schedules. We have very little time and energy left in our hands by the end of a busy day to enjoy a whole movie or binge a TV series.

Also, the popularity of bite-sized content and short-form videos is at its peak and is ever-growing. The introduction and rise of platforms like TikTok, Instagram Reels, and YouTube Shorts have successfully modified people's preferences regarding the length of videos they watch.[1] They have been so successful at their plan that according to studies, the attention span of youth consuming short-form videos has reduced to the level of a goldfish.

So, to cater to this new preference and make watching longer videos more efficient, we intend to summarize the long videos into shorter ones. Doing so allows us to utilize our time more efficiently by only watching the essential parts of a video and helping consume as much content as possible in a short time.

1.2 Aims and Objectives

We aim to cater to the audience with a good user experience when consuming content. Therefore, we want to formulate something to help them consume a more efficient and summarized form of the initial content.
Video summarization aims to summarize an extensive collection of video data and achieve efficient access and representation of that video content. By watching the

summary, users can make quick decisions on the usefulness of the video and watch a rather lengthy video in the shortest amount of time possible without missing out on the story or its essence.

A few of our main objectives regarding this project are mentioned below:

- To detect key frames from a video
- To calculate the similarity between frames of a video
- To create clusters by K means clustering and Principal Component Analysis (PCA)
- To extract key frames successfully
- Compile the summarized video
- To create a concise and complete synopsis by selecting the most informative parts of the video content
- To create a temporally consistent summary

Chapter 2

Related Works

Similar works have been done by other authors for both key frame extraction methods and video summarization. The hardest part of the video summary is figuring down what is significant and what isn't and then separating the two. Low-level features like texture [2], shape [3], or motion [4] might be used to categorize the significant content. The summary is made by combining the frames that include this crucial data. The process of extracting essential details from static frames is known as key frame extraction. These techniques are mostly used to derive a static synopsis of the video. The most commonly used key frame extraction techniques are egocentric video summarization [5] and discrete cosine transform[6]

Key-frame extraction algorithms have been widely used in video retrieval, which is utilized for video browsing and content-based retrieval applications. Similar to how different key-frame extraction strategies have been put forth for motion capture data in recent years. Based on the domain of the key-frame extraction problem, these key-frame extraction techniques can be categorized into the following three groups: curve simplification, clustering, and matrix factorization.

Curve Simplification: Lim and Thalmann completed the initial step of curve simplification [7]. They used Lowe's curve simplification approach to extract key-frames in response to a performer's changes in position, treating the motion sequence as a trajectory curve in the high-dimensional feature space [8]. The connections between streamlined curve segments are where the keyframes are extracted. Together, Li et al. [9] and Xiao et al. [10] used a brand-new layered curve simplification approach for motikim2014jointon capture data [11]. Curve saliency for motion curves was employed by Clifford and Baciu [12], and Halit and Capin [13] to identify the crucial motion frames [12]. The kind of source data used in these procedures is different from one another.

Clustering: The clustering problem of key-frame extraction entails grouping frames having a similar posture. Key-frame extraction using adaptive clustering was carried out by Liu et al. [14]. A measure of similarity between the two frames was established. Using the specified similarity, they assigned each frame to a corresponding cluster. Quaternions were utilized by Park, and Shin [15] to represent motion data, while PCA and k-means clustering were used to linearize and cluster the data. Then, they extracted key-frames from clustered motion data using dispersed data interpolation. More recently, Phillips et al. [16] developed overlapping clustering based on network structure analysis. Takaki et al. [17] applied an underlying fuzzy clustering algorithm within the IGSCR framework to examine the consequences of using the fuzzy clustering method. They all achieved successful outcomes.

Matrix factorization: A matrix created by aligning each frame's vertices in a row served as the representation for an animated sequence in this category [18]. A weight and key-frame matrix were then roughly formed from this matrix. Then, using methods like low-order discrete cosine terms (DCT) and singular value decomposition (SVD), the summary of motion was created [19] [20].

These techniques combine clustering techniques with low-level characteristics and dissimilarity detection to extract static keyframes from a video. The valuable features to be included in the summary are extracted using clustering techniques, while irrelevant frames rich in low-level features are removed. Researchers have utilized a variety of clustering techniques to identify intriguing frames. To extract the keyframes, specific techniques, like those described in [1] and [21], use web-based picture priors.

We have focused mainly on the ResNet50 network, a popular CNN architecture as our pre-trained model in Transfer Learning.[22] Additionally, several Transfer Learning architectures were experimented with a few other popular pre-trained models (VGG16, VGG19, AlexNet) and compared with the proposed model. The proposed model has given the best performance of 99.80 percent training accuracy.[23] A system that intelligently detects a human from an image or a video is a challenging task in the modern era. Over the last decade, the computer vision and pattern recognition community concentrated on human detection largely due to the variety of industrial applications, which include video surveillance, traffic surveillance, human-computer interaction, automotive safety, real-time tracking, human-robot interaction, search and rescue missions, humans' collective behavior analysis, anti-terrorist applications, pedestrian detection, etc. Although these criteria are required by these algorithms, they do not apply to all experimental data. It is exceedingly challenging to choose experimental settings that can affect the number of clusters while analyzing a movement without knowing the motion's content. Incorrect initial values significantly impact the outcomes of experiments. Setting a standard threshold for all experimental data and obtaining reliable results is impossible due to the variations in movement sequence duration and motion type.

In this project, we use a pre-trained ResNet50 model that extracts the feature vectors from an input video and reduces the dimensions of the vectors using PCA. After that, we cluster them using K-means clustering, and the cluster centroids are identified as keyframes. Then we compile the frames to get our desired summarization.

Chapter 3

Background and Design of the System

A basic video summarizing system extracts the image features from the video frames and then chooses the most representative frames by examining the visual variations between the visual features.

A holistic view of the entire movie or the local differences between neighboring frames is used to achieve this. Most approaches rely on universal characteristics like color, texture, motion data, etc. Summarization methods also use clustering. Two types of video summarization can be characterized:

1. Keyframing for static video summaries and
2. Dynamic video summaries (video skimming)

Dynamic video summaries are made up of a series of shots and are formed by considering the similarity or domain-specific relationships among all video shots as opposed to static video summaries, which are made up of a set of keyframes taken from the original video.

One benefit of a video skim over a keyframe set is the ability to incorporate audio and motion components that may improve both the expressiveness and the amount of information the summary conveys. In addition, seeing a skim rather than a slide presentation of keyframes is frequently more enjoyable and intriguing.

As opposed to a strict sequential display of video skims, keyframe sets give much more freedom in terms of organizing for browsing and navigation. This is because they are not constrained by time or synchronization difficulties.

3.1 Analysis of the Design Principles

At first, we need to split the video into frames; to do that, we have to extract frames at a desired rate from the video. After the extraction is complete, we load a pre-trained CNN, i.e., the ResNet50 model. After a frame has been trained on the ResNet-50 Model to extract features from it, we omitted the classification block. Then, we remove the last layer of the model named 'Predictions,' which enables us to extract a feature vector because it's the pooling layer. Concisely, we collect frames from a video, and then to collect features, we are using ResNet50.

After generating feature vectors, we also generate feature maps of the frames. Then, we calculate the Euclidean distance between two feature vectors. We calculated distances between frames with time differences of 1,2,3,4 and respective milliseconds. By doing this, we realized that the euclidean distance is proportional to the time difference between frames. i.e., the further the frame is, the larger the euclidean distance it will produce. This will help us to determine the highly familiar frames in the video, and we can decide which frames we need to keep for the video summary. We also used Principle Component Analysis (PCA) and K-means clustering to shorten the number of frames further.

Our initial approach to the design system for this project is explained below:

- **Extracting Frames:** A video is usually made up of a series of images called frames. Using a library from Python named **OpenCV**, we will be extracting **4 frames** per second. Using a loop method, we set our range to take steps on each frame and save the frame accordingly. All the frames are saved in a new folder.

- **Extracting Features from Frames:** We loaded a pre-trained Convolutional Neural Network (CNN), the ResNet50 model. To extract the features, we omitted the classification block. After that, we remove the last layer of the model to extract a feature vector because it's the pooling layer. In short, we are collecting feature vectors of frames from a video.

- **Calculating Euclidean Distance:** The resulting frame is analyzed to obtain the feature frame matrix. Every image has RBG associated with each of its pixels. The first is taken as a reference frame. The word count for each pixel value is taken in a vector form i.e. the vector represents a single frame, and the whole video is represented in a set of vectors.

  The distance between this frame is calculated using Euclidean distance. Where

$$distance = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

Figure 3.1: Euclidean Distance Formula

  x and y represent two frames of an image, and i represent columns. After the Euclidean distance between two consecutive images is calculated, a threshold value is given; when a distance exceeds a given threshold, a keyframe is claimed, and that frame serves as a new reference frame.

- **Clustering and Reducing Frames:** The extracted features of the frame are clustered (using K-means or Gaussian clustering) and are classified into different classes based on the distance calculated using the Euclidean distance measure. Each class is classified under the same frame name. The individual frame that has been extracted is considered a keyframe which, when combined, will compose a summary of a video.

- **Compiling the Video:** Segment the keyframes into shots and give each shot a shot-level significance score to create the video summaries from those shots. The sequences are then divided into video shots using K-means clustering, which also functions as a shot detection method, to create the final video summary.

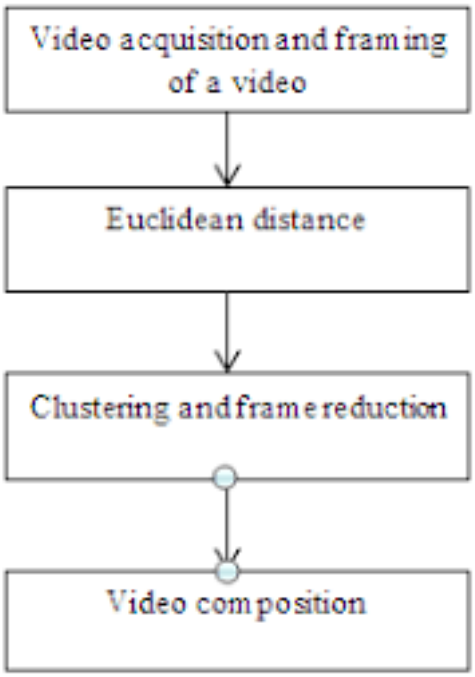


Figure 3.2: Framework

3.2 Usability, Manufacturing, Sustainability

Usability Testing Parameters

- Effectivity
- Efficiency
- Consistency

Effectivity: We should keep the major and important moments of a video to keep the original theme in the summary.

Efficiency: We should keep the time to generate the summarized videos short, so the main purpose of the system is fulfilled.

Consistency: We should control the summary that will comply with the expectations of the user in every aspect by following consistency in showing accuracy.

User Experience should include these three things for swift browsing:

- Reasonable
- Useful
- Desirable

Reasonable: We should get a short summary from a long video without a high processor is beneficial and also reasonable.

Useful: As time is a factor, it is very effective to give users proper information in a short time.

Desirable Nowadays short form videos are running in popularity. This allows the project to be desirable in the current context.

Chapter 4

Implementation of the System

Initially, we wanted to build our system in different phases where each will fulfill our various goals—starting from extracting scene images to shortening the video and then generating a story that concisely depicts the video. We will also be finding how accurate the story that we generated is in terms of accuracy.

- **Frame Extraction:** A video is usually made up of a series of images called frames. Using a library from Python named **MoviePy**, we will be extracting 4 frames per **s**econd. Using a loop method, we set our range to take steps on each frame and save the frame accordingly. All the frames are saved in a new folder.

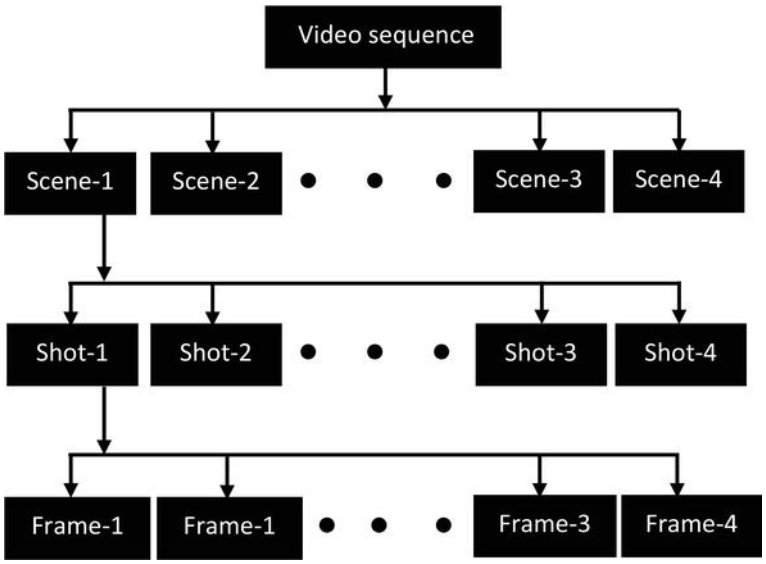


Figure 4.1: Frame extraction from videos

- **Feature Vector Extraction from Frames:** We loaded a pre-trained Convolutional Neural Network (CNN), the ResNet50 model. To extract the features, we omitted the classification block. After that, we remove the last layer of the model to extract a feature vector because it's the pooling layer. In short, we are collecting frames from a video.

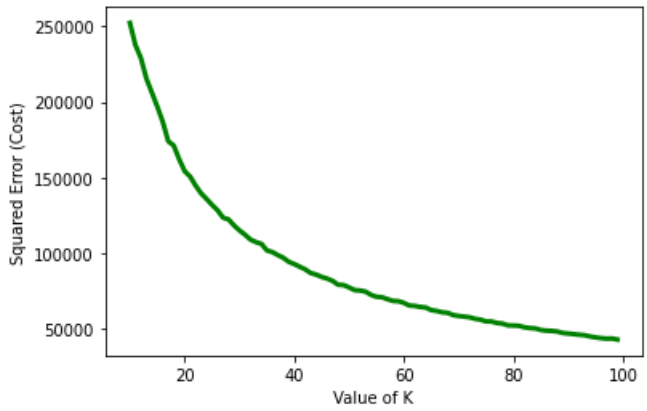
```
[ ]  key_frames[50]

     array([[[ 85,  72,  65],
             [ 89,  76,  69],
             [ 89,  80,  76],
             ...,
             [125, 123, 124],
             [125, 123, 124],
             [125, 123, 124]],

            [[ 85,  72,  65],
             [ 89,  76,  69],
             [ 89,  80,  76],
             ...,
             [125, 123, 124],
             [125, 123, 124],
             [125, 123, 124]],

            [[ 85,  72,  67],
             [ 89,  76,  71],
             [ 89,  80,  76],
             ...,
             [125, 122, 126],
             [125, 122, 126],
             [125, 122, 126]],

            ...,

            [[105,  87,  66],
             [105,  87,  66],
             [104,  86,  65],
             ...,
             [ 46,  47,  35],
             [ 46,  47,  35],
             [ 46,  47,  35]],

            [[107,  89,  68],
             [104,  86,  65],
             [103,  85,  64],
             ...,
             [ 44,  47,  35],
             [ 44,  47,  35],
             [ 44,  47,  35]],
```

Figure 4.2: NumPy Array Visualization of a key frame

- **Dimension Reduction and Clustering:** After extracting the feature vectors, we reduce their dimensions using Principle Component Analysis (PCA). Then we cluster them using K-means clustering, which is classified into different classes based on the distance calculated using the Euclidean distance measure. Each class is classified under the same frame name. The individual frame that has been extracted is considered a keyframe which, when combined, will compose a summary of a video. Thus, The keyframes are the cluster centroids. Here, we generate a Mean Squared Error graph to determine the number of clusters we should choose for the most optimal results for each unique video and the optimal score for the clusters. Principal Component Analysis is a feature extraction technique that maps a higher-dimensional feature space to a lower-dimensional feature space. While reducing the number of dimensions, PCA ensures that maximum information of the original dataset is retained in the dataset with the reduced no. of dimensions, and the co-relation between the newly obtained Principal Components is minimum. The new features obtained after applying PCA are called Principal Components and are denoted as PCi (i=1,2,3…n). Here, (Principal Component-1) PC1 captures the maximum information of the original dataset, followed by PC2, PC3, and so on.
  Our clustering algorithm consists of selecting k random centroid points on our multi-dimensional space and computing each distance against the clustered centroids. Each distance is assigned to the cluster for noise reduction and recomputed. On every iteration, we see if the centroids are converged; if not, we compute again.

```
[ ]  cost = []
     for i in range(10,100):
       KM  =  KMeans(n_clusters=i,max_iter = 500)
       KM.fit(X_pca)
       cost.append(KM.inertia_)
```

```
[ ]  plt.plot(range(10,100),cost,color='g',linewidth='3')
     plt.yscale('linear')
     plt.xlabel("Value of K")
     plt.ylabel("Squared Error (Cost)")
     plt.show()
```




```
[ ]  n_clusters = 98
     score = 2500000
```

```
[ ]  KM = KMeans(n_clusters=n_clusters,max_iter=500)
     KM.fit(X_pca)
     cost.append(KM.inertia_)
     cluster_centers = KM.cluster_centers_
     labels = KM.labels_
```

Figure 4.3: Code for K-means Clustering

- **Summarized Video:** Finally, we combine the frames using the OpenCV module of Python and get our desired summarized output video.

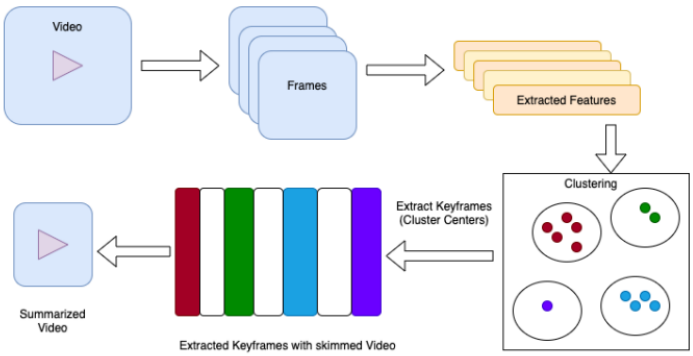
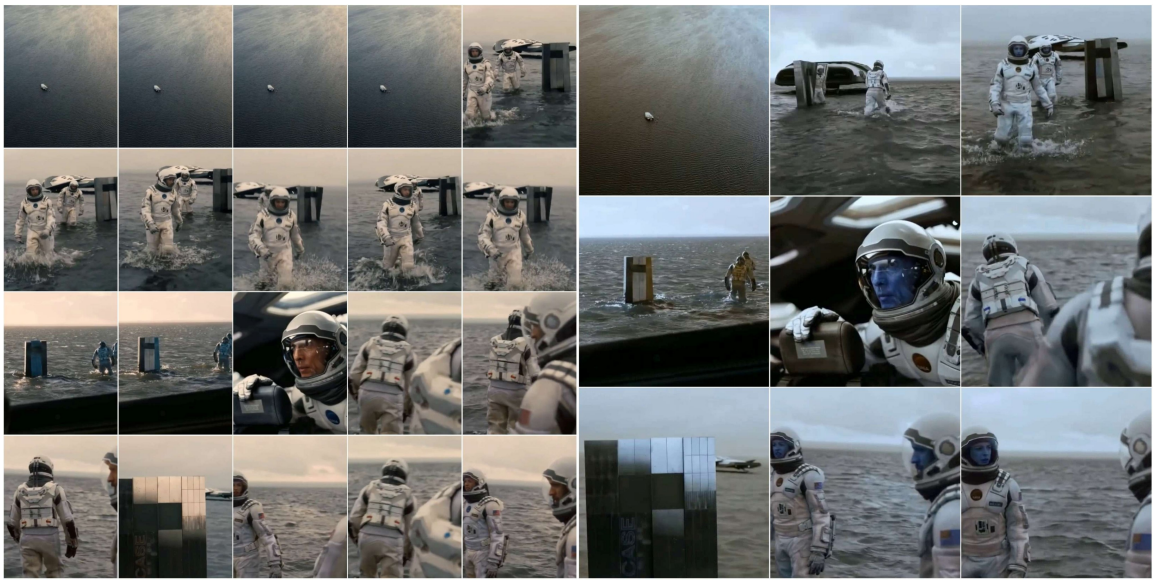


Figure 4.4: Video Summarization Sequence

Below you can see that we have snapshots of our input video after each second and after that, you can see the keyframes our cluster centroid has generated from the input video.



Input Video Sequence Output Video Sequence

Figure 4.5: Frame Extraction from Input Video

Chapter 5

Process of the Development

For this project, we went for the Agile methodology because, with Agile software development, we can quickly adapt to requirements changes without negatively impacting release dates. Not only that, Agile helps reduce technical debt, improve customer satisfaction and deliver a higher quality product.
Agile is an umbrella term for several methods and practices.

We went for the Scrum methodology among all. Agile software development methodologies are iterative, meaning the work is divided into iterations, which are called Sprints in the case of Scrum. In Scrum, projects are divided into cycles (typically two or 3-week cycles) called Sprints. The Sprint represents a time box within which a set of features must be developed. Specific ceremonies such as the Daily Stand-up meetings, the Sprint Review Meeting, and the Demo characterizes the Scrum method. These meetings helped us to communicate swiftly with the team members and keep track of what we achieved to complete.

Below, you can see a graphical representation of our planning for the whole semester that we achieved each week.

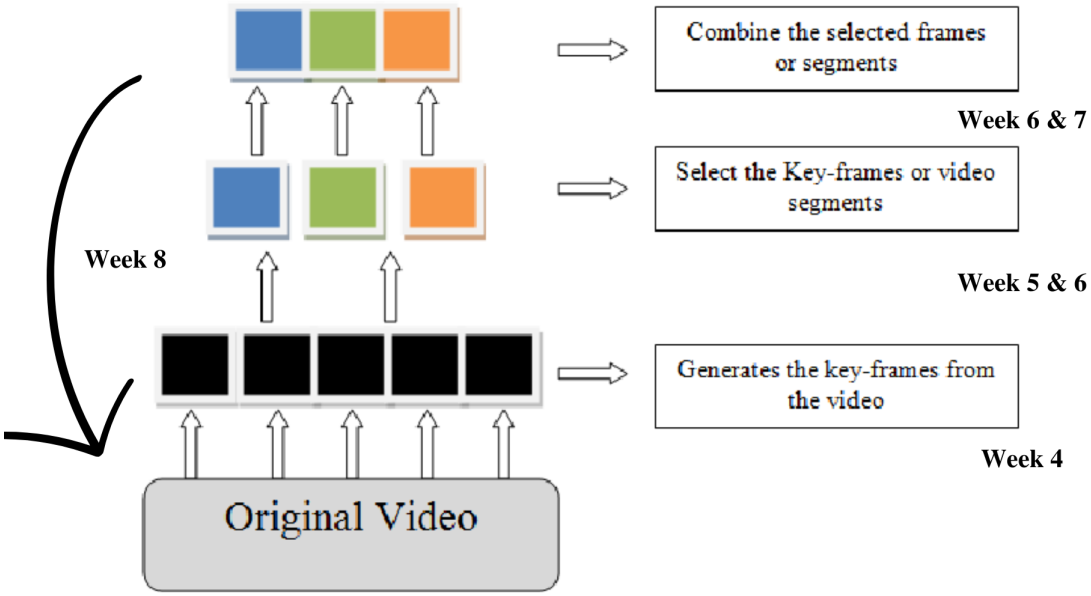


Figure 5.1: Work distribution

Project Management Method: **Agile**

We are going for the top down approach regarding this project. Instead of approaching software from scratch, we are breaking it into smaller components and dividing among the team members.

In order to create our agile pipeline for this project, we followed the scrum methodology and divided our work so that it is much more flexible, collaborative, efficient, and iterative.

Weeks/ Project Phases	1	2	3	4	5	6	7	8
Technical requirement findings & planning	■	■	■					
Implementing CNN model				■	■			
Finishing product backlog and sprints						■	■	
Test & Delivery								■

Figure 5.2: Work distribution

Chapter 6

Economic, Social, Political, Health Impact

6.1 Economic Impact

Getting a proper summary of a video requires highly efficient models implemented, which turns out to be a pretty resource and time-consuming task. So, we are trying to cut down that cost so that anyone can summarize videos accurately without having access to high-processor machines.

6.2 Social Impact

According to Jiang, Sadka, and Crookes [24], video summaries should adhere to the following three principles: priority, lack of redundancy, and continuity. Priority indicates that the high-priority items or events should be included in the output video summary. The final summary's non-repetitive keyframes are referred to as being "free of redundancy." Continuity refers to the output video summary's fluency; the important frames produced should match the temporal sequence in order to maintain the information's original significance. The use of video summarization in other contexts includes video surveillance [25] [1], event summarization [26], multimodal topic detection [27], and video-based question answering [28]. In today's context,

it is arduous for someone to efficiently navigate an endless collection of videos that caters to their tastes and preferences. An effective video summarizing facilitates viewers' browsing and navigation in large video collections, thus increasing viewers' engagement and content consumption.

6.3 Political Impact

Our project does not have any impact on any political issue.

6.4 Health Impact

We spend hours and hours staring at our screens [29]. More often than not, we find videos that feel mundane and uninteresting. According to studies, screen usage negatively influences our physical and emotional well-being and children's development. A sedentary lifestyle with little to no physical activity is the effect of excessive screen usage. Screen artificial light impacts our eyes, brain, and sleep. In response to these worries, experts advise limiting screen usage. Significant progress has been made in recent years to limit screen time. Some governments have imposed regulations on its use, and some businesses have considered enabling workers to take breaks from work to avoid screens. Screen time is a significant risk factor or a marker of mental disorders among US adults. Since mental health is predicted to be the leading cause of disease burden by 2030, the intervention should be targeted toward preventing these risk factors. [30]

So, how much screen time is recommended? Although it may seem like a straightforward question, the "just right" quantity frequently depends on the sort of screen a person is seeing and the purpose for doing so. Adults should make an effort to restrict their screen usage after work.

Video summaries will be handy here as they won't take up much of your time. This will allow the eyes of the users to rest without missing out on entertainment.

Chapter 7

Environment Consideration and Sustainability

-The sustainability of a project refers to the use of wasted resources, carbon footprint emission, usability issues, quality, and energy consumption. Sustainability in engineering is primarily an approach to design, implementation, and development that emphasizes energy efficiency and sustainability.

Nowadays, global issue promotes competition and force companies to implement energy and energy-efficient technology services. Whatever our work is, we have to think of a world that should be green and inhabitable.

Carbon emissions are measured as a carbon footprint (CF). It defines how much carbon dioxide a product or service produces in a particular cycle. Over the years, the IT sector has accounted for more than 2% of the world's CO2 emissions [31]. Any percentage of carbon dioxide emission is harmful to the environment. So it is crucial to think about the CF and the overall idea, development, delivery, and usage of the project. Climate pledges are the most vital part of being noted first. From the idea to handover any project, in every step, CF is calculated. We can't cut down the CF entirely, but we can minimize it in every step. Also, it is to be noted

that the actual calculations for the CF value of an AI model can not be obtained as of yet since the models used for the calculations are still in the early stages of development[32].

Usage is one of the longest periods of any software life cycle. At this time, the project is executed in a set of computers. As it is an abstract entity, the program doesn't directly create CF. But each cycle requires energy to execute. In many cases counting the used CPU time directly is very difficult it can only be estimated with certain approximations.

Usability issues and wasted resources are also crucial for any project. If the project leads a user to an undesirable outcome, it keeps the user in an incomplete phase, and if the resource is wasted, somehow, it maximizes the total cost.

The printing output is considered the most potential CF stage of any project. In this stage, the whole output is printed containing carbon. This used carbon is also harmful to the environment.

Sustainable development practices are nowadays very important and appreciated worldwide. In the realm of AI, 'ResponsibleAI' is gaining more traction now because of the focus on CF of AI models. Customers are likely to appreciate these efforts to minimize environmental harm. Thus the projects are becoming more efficient and ecologically and socially responsible.

A study discusses how we can make ML more efficient, the study claims that by following these 4 techniques, we can reduce energy consumption by up to 100 times and carbon emissions by up to 1000 times compared to following orthodox choices[33]. They coined these techniques as the "4Ms". The 4 M's to reduce energy consumption by an ML model are noted below:

- Model: Selecting efficient ML model architectures while advancing ML quality, such as sparse models versus dense modes, can reduce computation by factors of 5–10.

- Machine: Using processors optimized for ML training, such as TPUs or recent GPUs (e.g., V100 or A100), versus general-purpose processors can improve performance/Watt by factors of 2–5.

- Mechanization: Computing in the Cloud rather than on-premise improves data center energy efficiency3, reducing energy costs by a factor of 1.4–2.

- Map: Cloud computing lets ML practitioners pick the location with the cleanest energy, further reducing the gross carbon footprint by factors of 5–105.

The formula used for calculating the energy consumption in the study is: MWh = (Hours to train X Number of Processors X Average Power per Processor) X PUE where MWh is Megawatt Hours to measure energy, PUE is Power Usage Effectiveness which is the industry standard metric of data center efficiency, defined as the ratio between total energy usage divided by the energy directly consumed by the datacenter's computing equipment.

Then the energy in the MWh unit is turned into carbon by multiplying it with the carbon intensity of the energy supply: tCO2e = MWh X tCO2 per MWh, where Carbon intensity (tCO2e per MWh) is a measure of the cleanliness of a datacenter's energy[33]

Our goal in this project is to summarize a long video into a shorter one using LSTM, and we want to vehemently try to implement these 4 techniques as much as possible in our project. Following these 4 techniques, we can evaluate our current model in the following way: As we are summarizing a long video into a shorter one without using any high processor, it will not give a load in storage, which is a good feature here.

Again if we think of any Machine Learning project, the most important things that come to mind are power, cost, cost minimization, environment creation, etc. We are trying to solve the summarization part without any usability issues. Again, the process of summarizing a long video is time-consuming in some ways. But it doesn't waste resources and power energy. We are also doing most of our computing required for the project in the cloud using Google Colaboratory, which evidently reduces energy costs, as presented in the study.

Our project also requires nothing to be printed, and we need no paper. Thus, holistically carbon dioxide emission is negligible. The power we need to summarize a video is not of high voltage, and we are also reducing power consumption by summarizing the long video into a shorter one. As we are not using a high processor, it is not costly as well any ordinary devices like computers and laptops in our day-to-day life can be used for this.

So the cost minimization factor works well here. The most exciting part of our project is that it has no printed output. Therefore our model is giving us an output without carbon emission at this stage which is one of the most efficient parts alongside it being computed on a cloud-based server. Our goal now, is to keep the CF factors in mind and code the whole project most efficiently and optimally possible for the environment and contribute even so to sustainability.

Chapter 8

Ethical and Professional Responsibility

Our main goal in ethical terms would be to ensure that copyrights are not violated while summarizing the video. Many a time these summarized videos can be used unethically which might lead to getting flagged by organizations on copyright infringement terms.

Different professions have many ethical problems, like conflicts of interest. However, developing machines whose external actions mimic human behaviors that we deem "intelligent" raises particular ethical challenges in computer science. We reevaluate their connection to the artifacts they create, develop, and ultimately deploy as machines become more diverse and intelligent and solely take on activities previously reserved for humans. That is, we checked the summaries manually to find out whether the summarized versions were free from flaws and errors.

We have given proper credit to the works we have taken inspiration from in our report. All the models and frameworks that we used are cited with fair acknowledgment and credit.

Chapter 9

Tools and Technology

9.1 Libraries

- Keras
- Tensorflow
- Numpy
- Pandas
- SciPy
- MoviePy
- OpenCV

9.2 Pre-trained CNN Model

- ResNet50

9.3 Code Editor

- Google Colaboratory
- Visual Studio Code

Chapter 10

Result Analysis

We applied the method to a lot of online videos downloaded from several video websites and cropped from movies and series. Our module was implemented in Python and OpenCV 2.0, and then the experiments were conducted on a Windows 10 system with an Intel i5 processor and 8 GB RAM.

Firstly, we took television show "Mad Men" to verify the effectiveness and robustness of the proposed method. More than 20 versions of the copies or near-duplicates were downloaded, which may be different in video formats (.mp4, .rm, .wmv, .flv, etc.), spatial resolutions (1920*1080, 1080*720, 720*576, etc.), video lengths (such as short clips cut from a full video), and so on. The results which are got from the downloaded video with mp4 format are partly shown in the Figures 10.1 and 10.2.

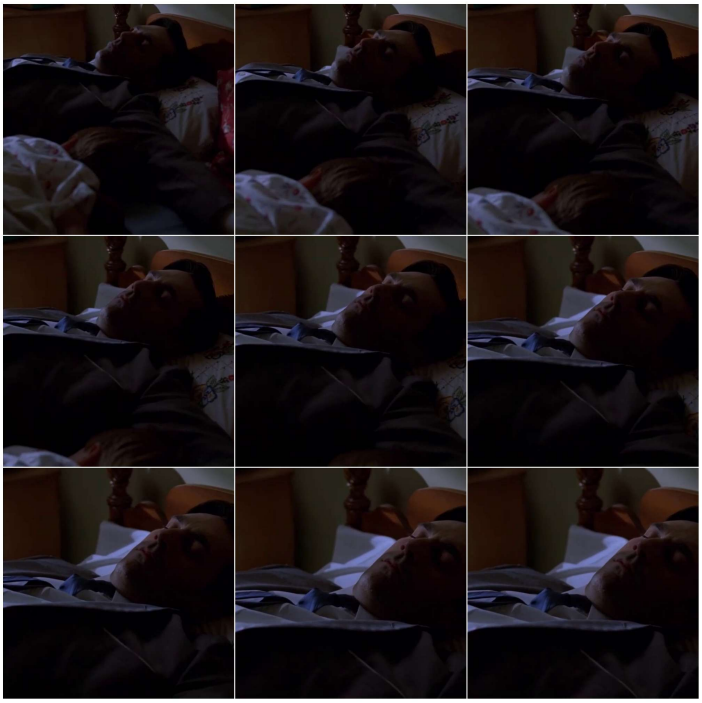


Figure 10.1: Input Video

From Figure 10.1, we can see that almost the same background and scene are in all the frames. The difference among these frames is the hand motion and camera angles. So the final key frames are extracted based on the structural difference from the alternative key frames, as shown in Figure 10.2. The frame content could be viewed definitely and their order consisted of the original video, and there is appropriate redundancy, such as the first three frames of the summarized video.

In short, we found out how different the motion in each frame is and extract the unique features using a CNN model. They were later clustered and went through PCA and K-means to categorize. After that, we combined them using the CV2 package of Python and got our desired summarized video. In some summaries, the feature vectors showed discrepancies and lacked RGB color orientation. For each different video the results that we got were different and varied with significant changes.

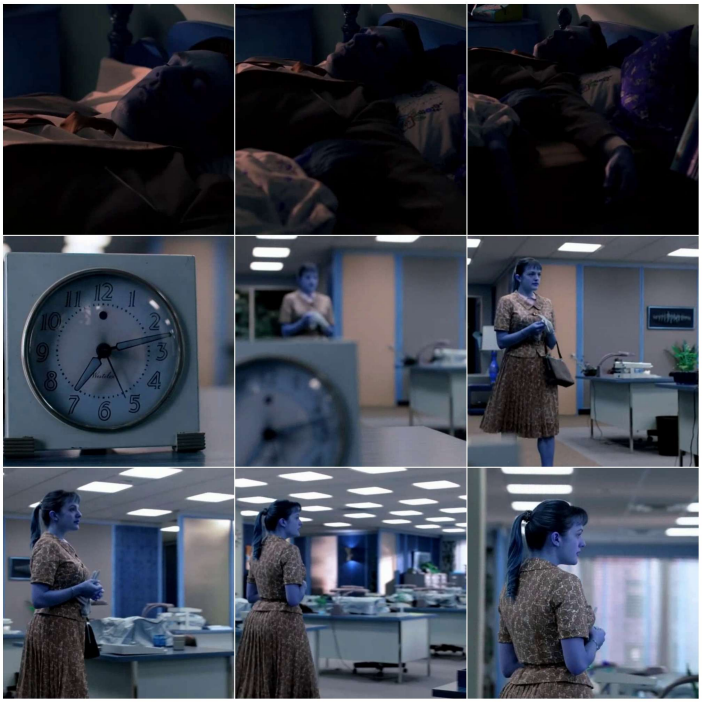


Figure 10.2: Final Video

Below you can see how at each timestamps, we get different frames for three kinds of videos.

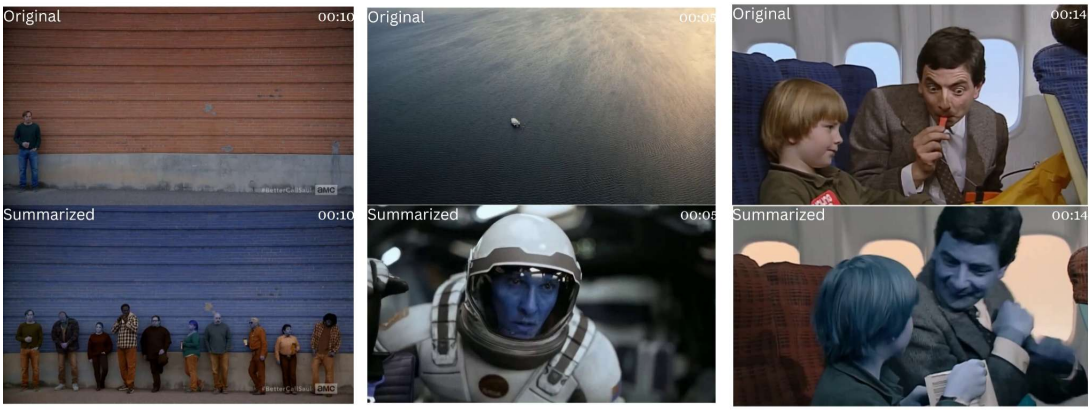


Figure 10.3: Video summarization

Here at each timestamp, we have tried to showcase what frame has been generated from our approach of video summarization that extracts features from a ResNet50 model and then goes through PCA for dimension reduction. We then cluster them using K-means clustering to identify the key frames.

Chapter 11

Conclusion

Video summarization is a rapidly expanding field that has applications in many different sectors. One of the most difficult tasks is video summarization because it depends on the individual. Therefore, we can never establish a reliable baseline to determine whether our algorithm is effective. Sometimes, humans only need 1-2 seconds of video to summarize something, but machines can give us up to 10 seconds of video by looking for even the smallest difference in image intensity.

ResNet50, PCA, and K-means clustering are used in this work to implement our approach. The feature extraction technique utilized here, ResNet-50, performs better than most similar techniques. Our approach also preserves the video summary's temporal consistency, resulting in summaries closer to the original video. Since the NumPy arrays are directly converted into the video, we couldn't keep the color accuracy of the generated videos.

So far, we have completed the video summarization module of the project as of now. Furthermore, we want to generate a story or textual analysis from the shortened video, which makes up our story generation module. After that, if we get enough responses, we could make its software and add more customizing tools to it.

11.1 Future Work

So far, our project has only been implemented on some test videos to generate the summarized video. If we could implement our approach and build a tool/software, then it would be very much in demand as short-form videos are now more popular than ever.

As for generating revenues, we could set customization features for the summarized video, like duration, frequency, specific object scenes, etc., and provide them as a premium feature and monetize it.

There are also research options open for this project, such as:

- Studying the effect of the size of the video to apply efficient video summarization by applying optimization methods to work for different video sizes, even for long size video.
- Studying the effect of quality of the video by summarization bad quality video.

Bibliography

[1] C. Wang, M. Gu, X. Wang, P. Ong, Q. Luo, and Y. Li, "Research on the challenge of the new short video platform tiktok on the traditional internet social media facebook," Sep. 2021.

[2] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[4] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International journal of computer vision*, vol. 61, no. 3, pp. 211–231, 2005.

[5] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 1346–1353.

[6] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34–44, 2013.

[7] I. S. Lim and D. Thalmann, "Key-posture extraction out of human motion data," in *2001 Conference Proceedings of the 23rd Annual International Con-*

ference of the IEEE Engineering in Medicine and Biology Society, IEEE, vol. 2, 2001, pp. 1167–1169.

[8] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[9] S. Li, M. Okuda, and S. Takahashi, "Embedded key-frame extraction for cg animation by frame decimation," in *2005 IEEE International Conference on Multimedia and Expo*, IEEE, 2005, pp. 1404–1407.

[10] H. Togawa and M. Okuda, "Position-based keyframe selection for human motion animation," in *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, IEEE, vol. 2, 2005, pp. 182–185.

[11] J. Xiao, Y. Zhuang, T. Yang, and F. Wu, "An efficient keyframe extraction from motion capture data," in *Computer Graphics International Conference*, Springer, 2006, pp. 494–501.

[12] K. S. Clifford and G. Baciu, "Entropy-based motion extraction for motion capture animation: Motion capture and retrieval," *Comput Animat Virt W*, vol. 16, no. 3-4, pp. 225–235, 2005.

[13] C. Halit and T. Capin, "Multiscale motion saliency for keyframe extraction from motion capture sequences," *Computer Animation and Virtual Worlds*, vol. 22, no. 1, pp. 3–14, 2011.

[14] F. Liu, Y. Zhuang, F. Wu, and Y. Pan, "3d motion retrieval with motion index tree," *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 265–284, 2003.

[15] M. J. Park and S. Y. Shin, "Example-based motion cloning," *Computer animation and virtual worlds*, vol. 15, no. 3-4, pp. 245–257, 2004.

[16] R. D. Phillips, L. T. Watson, and R. H. Wynne, "A study of fuzzy clustering within the igscr framework," in *Proceedings of the 46th Annual Southeast Regional Conference on XX*, 2008, pp. 140–145.

[17] M. Takaki, K. Tamura, Y. Mori, and H. Kitakami, "A extraction method of overlapping cluster based on network structure analysis," in *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, IEEE, 2007, pp. 212–216.

[18] K.-S. Huang, C.-F. Chang, Y.-Y. Hsu, and S.-N. Yang, "Key probe: A technique for animation keyframe extraction," *The Visual Computer*, vol. 21, no. 8, pp. 532–541, 2005.

[19] M. Cooper and J. Foote, "Summarizing video using non-negative similarity matrix factorization," in *2002 IEEE Workshop on Multimedia Signal Processing.*, IEEE, 2002, pp. 25–28.

[20] Y. Gong and X. Liu, "Video summarization using singular value decomposition," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, IEEE, vol. 2, 2000, pp. 174–180.

[21] G. Kim, L. Sigal, and E. P. Xing, "Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4225–4232.

[22] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[23] I. Z. Mukti and D. Biswas, "Transfer learning based plant diseases detection using resnet50," in *2019 4th International conference on electrical information and communication technology (EICT)*, IEEE, 2019, pp. 1–6.

[24] R. M. Jiang, A. H. Sadka, and D. Crookes, "Advances in video summarization and skimming," in *Recent advances in multimedia signal processing and communications*, Springer, 2009, pp. 27–50.

[25] W. Feng, D. Ji, Y. Wang, S. Chang, H. Ren, and W. Gan, "Challenges on large scale surveillance video analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 69–76.

[26] Z. Wang, P. Cui, L. Xie, H. Chen, W. Zhu, and S. Yang, "Analyzing social media via event facets," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 1359–1360.

[27] V. Chu, K. Bullard, and A. L. Thomaz, "Multimodal real-time contingency detection for hri," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 3327–3332.

[28] L. Zhu, Z. Xu, Y. Yang, and A. G. Hauptmann, "Uncovering the temporal context for video question answering," *International Journal of Computer Vision*, vol. 124, no. 3, pp. 409–421, 2017.

[29] N. Stiglic and R. M. Viner, "Effects of screentime on the health and well-being of children and adolescents: A systematic review of reviews," *BMJ open*, vol. 9, no. 1, e023191, 2019.

[30] K. Madhav, S. P. Sherchand, and S. Sherchan, "Association between screen time and depression among us adults," *Preventive medicine reports*, vol. 8, pp. 67–71, 2017.

[31] N. Jones, "How to stop data centres from gobbling up the world's electricity," *Nature*, vol. 561, no. 7722, pp. 163–167, 2018.

[32] N. Bannour, S. Ghannay, A. Névéol, and A.-L. Ligozat, "Evaluating the carbon footprint of nlp methods: A survey and analysis of existing tools," in *EMNLP, Workshop SustaiNLP*, 2021.

[33] D. Patterson, J. Gonzalez, U. Hölzle, *et al.*, "The carbon footprint of machine learning training will plateau, then shrink," *Computer*, vol. 55, no. 7, pp. 18–28, 2022.